REMARKS

[0002] Applicant respectfully requests reconsideration and allowance of all of the claims of the application. Claims 1-28, 34-42, and 45-46 are presently pending. Claims amended herein are 1, 11-13, 26-28, 34, 38-40, and 45. Claims withdrawn or cancelled herein are 29-33, 43, 44, and 47-50.

Formal Request for an Interview

[0003] If the Examiner's reply to this communication is anything other than allowance of all pending claims, then I formally request an interview with the Examiner. I encourage the Examiner to call me—the undersigned representative for the Applicant—so that we can talk about this matter so as to resolve any outstanding issues quickly and efficiently over the phone.

[0004] Please contact me or my assistant to schedule a date and time for a telephone interview that is most convenient for both of us. While email works great for us, I welcome your call to either of us as well. Our contact information may be found on the last page of this response.

Claim Amendments

[0005] Without conceding the propriety of the rejections herein and in the interest of expediting prosecution, Applicant amends claims 1, 11-13, 26-28, 34, 38-40, and 45 herein. Claims 47-50 are cancelled without prejudice or disclaimer.

[0006] Claim 1 is amended to recite, *inter alia*, "[a]n interceptor configured to intercept non-native kernel calls that call a native kernel from non-native program

modules, the native kernel being software that operates system functions." Support for the amendment can be found throughout the Application including, for example, Fig. 3 with the associated text.

[0007] The kernel, referred to in the Application, is described as follows:

> The next layer of the architecture is the kernel mode 130. This may be generally called the "kernel" of the OS. Since it is part of the OS, it is part of the computing platform.
> A kernel of an OS is the privileged part of the OS—the most trusted part of the OS. It is an inner layer of code. It typically operates I/O 132, security 134, display control (i.e., access to the screen) 136, memory management 138, and other privileged functions 139. The kernel has sole access to the hardware in the hardware layer 150 via device drivers 142 and other hardware interfaces 144. (Specification at p.4, line 8-15).

[0008] Claim 1 is also amended to recite, *inter alia*, "an I/O unit configured to deliver the native kernel calls converted by the call-converter to the native kernel…" Support for the amendment can be found throughout the Application including, for example, Fig. 4 with the associated text.

[0009] Claims 13, 34, 40, and 45 are similarly amended, and therefore are supported by the Application too.

[0010] Accordingly, no new matter will be added by the amendment. Entry to the file is respectfully requested.

Substantive Matters

Claim Rejections under § 101

[0011] Claims 1-10, 40 and 41 are rejected under 35 U.S.C. 101 for being software and *per se* non-statutory subject matter. (03/28/2007 Office Action at p.2).

[0012] Claim 1 is amended to recite "[a] kernel emulator implemented at least in part by a computing device for non-native program modules, the kernel emulator comprising..." The Examiner is directed to many issued patents, in which claims written to incorporate "[s]oftware architecture" implemented by a computing device are considered judicially-created subject matter, and therefore are allowed by the Patent Office. For example, claim 14 of U.S. Patent No. 7,107,577 to de Boer et al. recites "[a] software architecture implemented in a computer for controlling an apparatus including one or more hardware units each for performing one or more corresponding functions, comprising..." Furthermore, claim 1 of U.S. Patent No. 7,165,239 to Hejlsberg et al. recites "[a] software architecture for a distributed computing system implemented at least in part by a computing device, the software architecture comprising..." Those claims are allowed by the Patent Office even though each of the subject matters involves software.

[0013] Following the examples above, since the kernel emulator is software, the amended claim 1 "[a] kernel emulator implemented at least in part by a computing device" shall overcome 35 U.S.C. 101 rejection too. Applicant respectfully requests withdrawal of rejections to claims 1-10, 40, and 41.

[0014] Claims 11, 12, 26-28, 38, and 39 are rejected under 35 U.S.C. § 101 for failing to produce tangible results. (03/28/2007 Office Action at p.3).

[0015] Claim 11 is amended to recite, *inter alia*, "whereby the calls from the non-native program modules are processed by the native kernel through the kernel emulator without modifying the non-native program modules."

[0016] Applicant respectfully submits that by this amendment to claim 11, a "useful, concrete and tangible" result is recited. The Examiner is directed to the Application, which is purported to facilitate the operation of non-native program modules within a native computing platform. (Specification at p.1, lines 8-9). Accordingly, with the help of the emulator addressed in the Application, older generations of applications may run on the latest generation of the OS. Therefore, end users who purchase the new OS are not forced to discard their current applications, thus protecting their investment while still benefit from features provided in the new OS. (Specification at p.4, line 21 to p.5, line 2). So, to run the non-native program on a native OS platform "without modifying the non-native program modules" is one of the "useful, concrete and tangible" results intended in the Application. Accordingly, amended claim 11 is asserted to overcome the rejection under 35 U.S.C. § 101.

[0017] Claims 12, 26-28, 38, and 39 are similarly amended, and therefore are asserted in compliance with 35 U.S.C. § 101 too.

[0018] If the Examiner maintains the rejection of these claims, then the Applicant requests additional guidance as to what is necessary to overcome the rejection.

Claim Rejections under § 102

[0019] Claims 13-25, 34-37 are rejected under 35 U.S.C. § 102 for being anticipated by U.S. Patent No. 5,623,617 to Davidian ("Davidian"). Applicant herein amends independent claims 13 and 34, rendering the rejections to these claims moot.

[0020] Independent claim 13, as amended, recites:

> 13. A method of emulating a kernel for non-native program modules, the method comprising:
>
> intercepting non-native kernel calls from non-native program modules, the non-native kernel calls calling a native kernel that comprises software and operates system functions;
>
> converting the intercepted non-native kernel calls into native kernel calls; and
>
> delivering the converted native kernel calls to the native kernel, whereby the non-native kernel calls from the non-native program modules are processed by the native kernel through the conversion without modifying the non-native program modules.

[0021] Applicant submits that Davidian fails to teach all the features in amended claim 13. In particular, Davidian does not teach to "intercept non-native kernel calls" that calls a native kernel, nor does it teach to "deliver the converted native kernel calls" to the native kernel for further processing.

[0022] Davidian is directed to a system for decoding guest instructions into host instruction by the host computer. Davidian teaches, *inter alia*, a decoding technique for emulating a guest program of instruction in a host processor through an optimized decoder. As shown in Fig. 1 below, the host processor system includes host addressable

memory, which includes an emulation program store 17, and a sequence of guest instructions in a guest program store 18. The emulation program store 17 includes an emulation dispatch store 20, which is an indexed table of pairs of host instructions corresponding to the entry points for emulation programs to emulate each of possible guest instruction encoding for a CPU-specific assembly language (i.e., Motorola 68020), and a set of emulation routines 21.

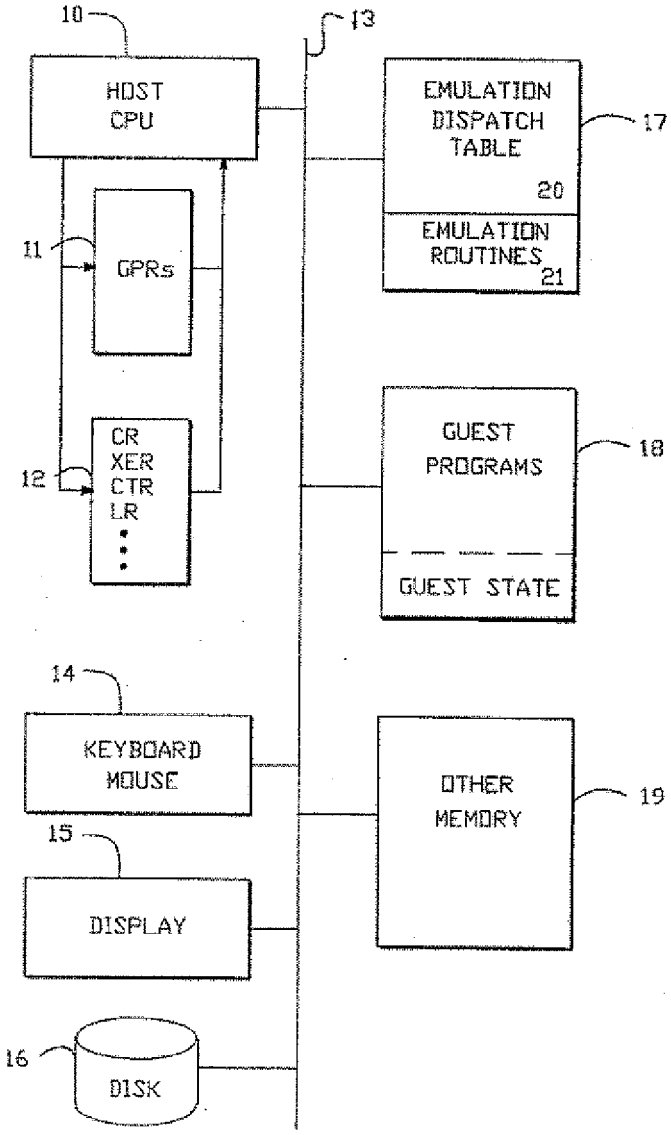

FIG. – 1

[0023] To achieve emulation, Davidian discloses first decoding the guest instructions into a sequence of host instructions which accomplish the result intended by the guest instruction, and then sending the decoded instruction to host processor 10 for execution.

[0024] Apparently, both guest instruction and decoded instruction target the host processor, rather than the native kernel enumerated in paragraph [0007]. According to Davidian, after decoding, instructions are directly dispatched to registers of the host processor. (See Fig. 1 of Davidian). In other words, the emulation method disclosed in Davidian only translates guest instructions to instructions native to the host processor such that the translated instructions are ready to be processed by the host processor.

[0025] The Examiner is directed to the difference between claim 13 and Davidian: (i) the non-native kernel calls that call kernel (software) from non-native program recited in amended claim 13 versus guest instructions that directly target processor (hardware) disclosed in Davidian; and (ii) after conversion, the delivery of the converted native kernel calls to the native kernel (software) recited in claim 13 versus dispatching of the decoded instructions to the host processor (hardware) in Davidian.

[0026] For the reasons above, claim 13 is respectfully asserted patentably distinct from Davidian. Claims 1, 34, 40, and 45 incorporate the same features, therefore, are asserted in condition for allowance.

Dependent Claims

[0027] In addition to its own merits, each dependent claim is allowable for the same reasons that its base claim is allowable. Applicant requests that the Examiner withdraw the rejection of each dependent claim where its base claim is allowable.

Conclusion

[0028] All pending claims are in condition for allowance. Applicant respectfully requests reconsideration and prompt issuance of the application. If any issues remain that prevent issuance of this application, the **Examiner is urged to contact me before issuing a subsequent Action**. Please call/email me or my assistant at your convenience.

Respectfully Submitted,

Dated: 07-30-2007

By: Ningning Xu

Ningning Xu
Reg. No. L0293
(509) 324-9256 x226
ningning@leehayes.com
www.leehayes.com

My Assistant: Carly Bokarica
(509) 324-9256 x264
carly@leehayes.com